Remarks/Arguments

With reference to the Final Office Action of April 30, 2007 and the Advisory Action mailed July 20, 2007, Applicants offer the following remarks.

Art Rejections

In the Final Office Action of April 30, 2007, claims 1-8, 10-18, 20-28, and 30 were rejected as being unpatentable over US Patent 6,125,444 to Check et al. for Millimode Capable Computer System Providing Global Branch History Table Disables And Separate Millicode Disables Which Enable Millicode Disable To Be Turned Off For Some Sections Of Code Execution But Not Disabled For All in view of Hennessy & Patterson, Computer Architecture: A Quantitative Approach.

Claims 11 and 21 are substantially parallel to claim 1, and all of the claims are dependent from either Claim 1 or Claim 11 or Claim 21. Thus the rejection and subsequent amendment of Claim 1 will be discussed in detail, with discussion applied to the other claims.

Check recites (as applied to claim 1):

> One reason that the millicode would need to control when instruction fetches can occur is times when the millicode is doing sensitive system operations that require control of the type of cache functions that can occur. There are two mechanisms that can be used to control branch history table actions. Which will be used is determined by the specific millicode sequence being invoked. At the time of the entry into a millicode routine certain millicode control registers are initialized to the desired values. One bit in one of these registers when initialized to a value of one will result in the matching mechanism of the branch history table to

be disabled and no further instruction fetches of the targets of these entries will be performed. (emphasis added) (Check, column 2, lines 28-40)

and

> In a processor such as in FIG. 1 that has a BHT (5) that is providing information that directs instruction fetching (2) the BHT functions in a nearly asynchronous fashion in order to predict the target instruction stream to have available to the instruction decode (3) the instruction text to provide instruction to the execution element(s) (4). The branch history table (5) is provided a starting address to search (6) which it uses to access the BHT array (8). It will increment (7) by an amount chosen in the design of the BHT to search for possible branch instructions. (Column 4, lines 21-30)

The Art of Record

United States Patent 6,125,444 to Mark Anthony Check, et al. for Millimode Capable Computer System Providing Global Branch History Table Disables And Separate Millicode Disables Which Enable Millicode Disable To Be Turned Off For Some Sections Of Code Execution But Not Disabled For All describes a millimode capable computer system that provides control to millicode to allow the BHT operations to continue except when special situations occur that require control of instruction fetch operations where the BHT can be turned off for some sections of code execution, but not disabled for all. A single free running BHT functions for both a normal mode and a millimode for the central processor which can execute in millimode with a branch history table directing instruction fetch for which both a global BHT disable and millicode disables exist. Hit detection logic receives input from the global BHT disable, as well as from an initialized control register bit and a processor control register bit to select the correct set target information and generate a "branch history table hit detected" control signal.

The cited portion of Patterson and Hennessey is a thorough discussion of branch target buffers and branch history tables in pipeline processors.

Applicants' Claimed Invention

Status of the Claims.

Claims 1-30 were originally presented for Examination.

All of the claims were rejected in the Office Action of September 20, 2006. Applicants amended their original claims to particularly point out and describe their invention, and distinguish over the art of record, while canceling Claims 9, 19, and 29.

Claims 1-8, 10-18, 20-28, and 30 are pending.

Exemplary Claim

Claim 1, as amended, is exemplary.

> 1. (Currently Amended) A method of operating a computer having a pipelined processor, comprising setting a bit within an instruction text field of a branch, said bit preventing the branch from being placed into a branch history buffer and into a branch target buffer to thereby prevent the branch from being written into the branch history buffer and branch target buffer and predicted and to make the branch only detectable at the time frame of decode.

Discussion: Art Rejections

As noted above, Claims 11 and 21 are substantially parallel to Claim 1, and all of the claims are dependent from either Claim 1 or Claim 11 or Claim 21. Thus the rejection and subsequent amendment of Claim 1 will be discussed in detail, with discussion applied to the other claims.

The overarching issue is whether the claims, as limited by the newly and previously added clauses and limitations are allowable over the art of record, with the added limitations related to

> setting a bit within an instruction text field of a branch, said bit preventing the branch from being placed into a branch history buffer and into a branch target buffer to thereby prevent the branch from being written into the branch history buffer and branch target buffer and predicted and to make the branch only detectable at the time frame of decode.

That is, the bit prevents the branch from being placed into both (1) a branch history buffer and (2) a branch target buffer. Writing this is recited in the amended claims to prevent the branch from being written into the branch history buffer and branch target buffer and (3) from being predicted, so that the branch is only detectable at the time frame of decode.

Claim 1 will be analyzed in detail.

Applicants' invention, as recited in the specification and claims, is a method, system, and program product for operating a computer having a pipelined processor. The first step is setting a bit within an instruction text field of a branch. Setting the bits prevents the branch from being placed into both a branch history buffer and a branch target buffer. This prevents the branch from being predicted and makes the branch only detectable at the time frame of decode.

Specific support for this limitation is in numbered paragraph [0017] of Published Patent Application 20005/0216713-A1,

> [0017] This is accomplished through a computer system, a method of operating a computer having a pipelined processor, and a computer program product for branch prediction in a pipelined CISC. This is implemented by defining a bit within an instruction text field of a branch whereby to prevent the branch from being placed into a branch target buffer and to thereby make the branch only detectable as the time frame of decode. This results in predicting the direction and target of a branch prior to decode, frequently using a branch prediction array (as a branch target buffer). The branch is tracked from the beginning of the pipe, decode, until the time frame that the given instruction is to be written into a branch prediction array. In carrying out the invention, the instruction text field may be denoted as a non-writable branch into the BTB. More particularly, the instruction field in the system area is denoted as a non-writable branch into the BTB in system so that the branch is blocked. The instruction field when denoted in the non-system area may encounter aliasing. As a general rule machine state altering code lies within an address range supported by branch tag bits of the branch target buffer. According to the invention branches which have targets that are highly non-constant can be blocked from branch predictions through the use the BTB blocking field in the instruction text. Also, state altering code in the system area can be denoted by a state bit within the BTB/BHT such that aliasing of branches within system area is prevented. (Page 2, ¶ [0017])

Claim 1, as originally presented, recited:

> 1. A method operating a computer having a pipelined processor, comprising defining a bit within an instruction text field of a branch whereby to prevent the branch from being placed into a branch target buffer to thereby make the branch only detectable as the time frame of decode.

In the original Office Action, original Claim 1 was rejected on Check, Figure 1 (showing a pipelined processor), i.e.,

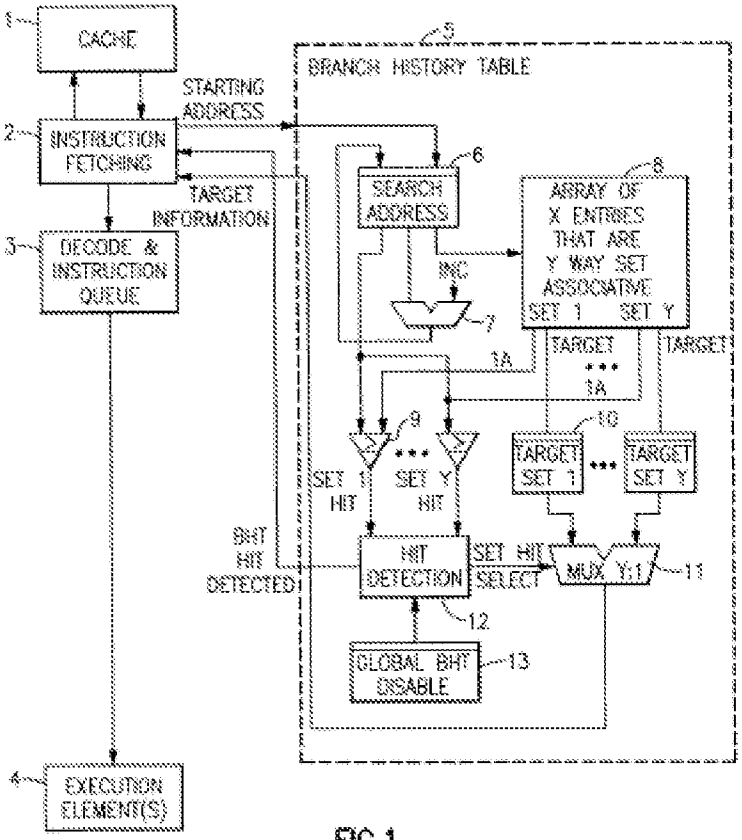


FIG.1

and Check column 2, lines 33-40-

> Which will be used is determined by the specific millicode sequence being invoked. At the time of the entry into a millicode routine certain millicode control registers are initialized to the desired values. One bit in one of these registers when initialized to a value of one will result in the matching mechanism of the branch history table to be disabled and no further instruction fetches of the targets of these entries will be performed. (Column 2, lines 32-40)

and Check, column 4, lines 24-27-

> In a processor such as in FIG. 1 that has a BHT (5) that is providing information that directs instruction fetching (2) the BHT functions in a nearly asynchronous fashion in order to predict the target instruction stream to have available to the instruction decode (3) the instruction text to provide instruction to the execution

> element(s) (4). The branch history table (5) is provided a starting address to search (6) which it uses to access the BHT array (8). It will increment (7) by an amount chosen in the design of the BHT to search for possible branch instructions. (Column 4, lines 22-31)

It is conceded in the original Office Action that while Check teaches using an instruction field of an instruction (actually, Check discloses "control registers"), Check does not disclose that doing so would prevent the branch from being placed in a branch target buffer. Note that this is positively recited in Claim 1, i.e.,

> "…said bit preventing the branch from being placed into a branch history buffer and a branch target buffer…"

It is further conceded in the Final Action that the claim limitation

> "said bit preventing the branch from being placed into a branch history buffer and a branch target buffer to thereby prevent the branch from being predicted and to make the branch only detectable ~~as~~ at the time frame of decode"

is not taught by Check, i.e., "While Check teaches using an instruction field of an instruction to disable a branch history table, he does not teach that doing so would prevent the branch from being placed in the branch target buffer."

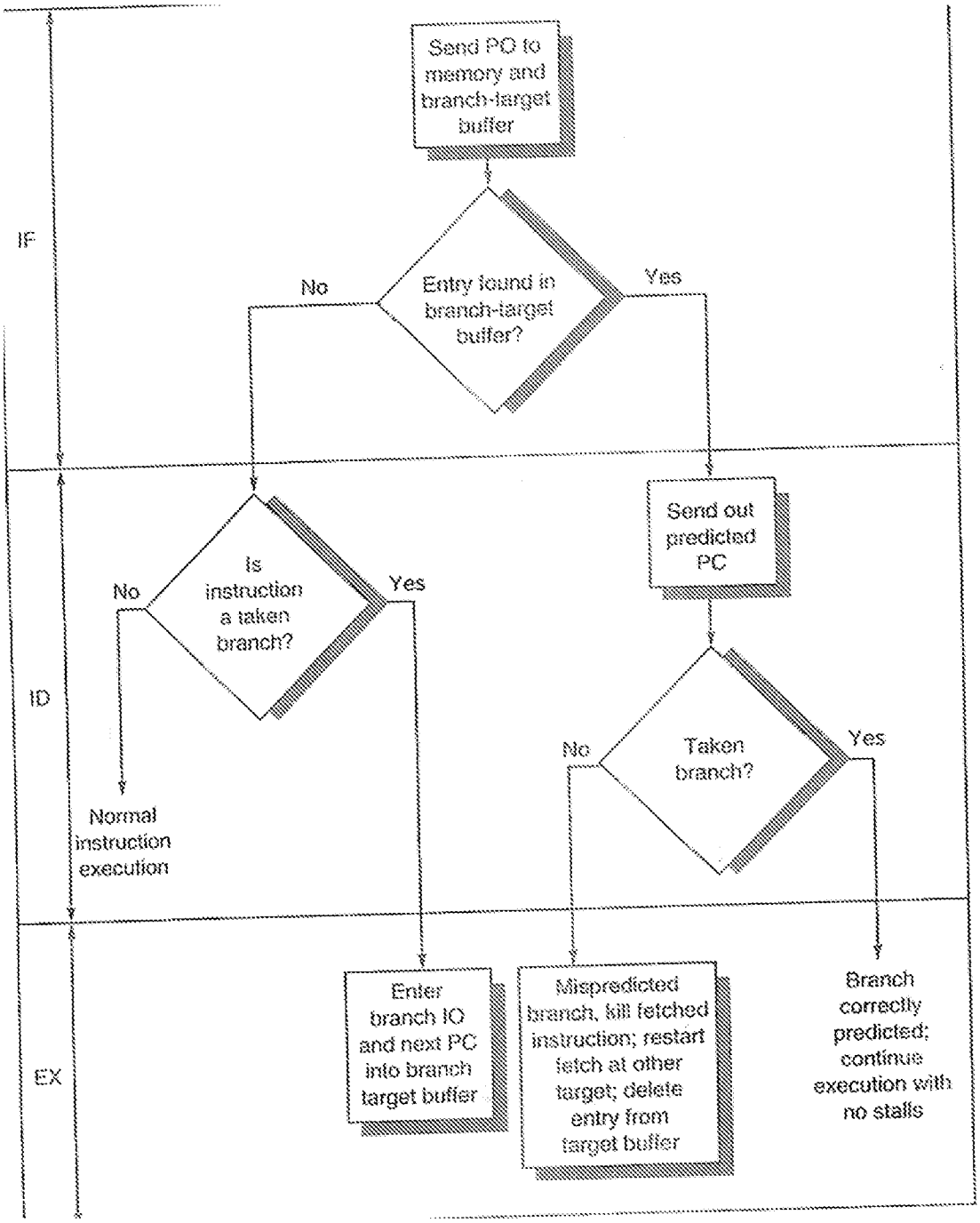
Note that this limitation has been amended as follows:

> setting a bit within an instruction text field of a branch, said bit preventing the branch from being placed into a branch history buffer and <u>into</u> a branch target buffer to thereby prevent the branch from being <u>written into the branch history buffer and branch target buffer and</u> predicted and to make the branch only detectable at the time frame of decode.

Patterson, pp. 271-275, is said to overcome this deficiency of Check, specifically because "As the branch target buffer relies on having a prediction in order to determine the correct address, as seen by Figure 4.23 on Page 276, if no prediction was able to be generated, because of a BHT being disabled, then it would have been obvious to one of ordinary skill in the art to also not use the BTB, as it would not have the data it needs to be made use of."

But, Patterson do not teach or disclose writing a bit to preclude generating a prediction or to preclude writing an entry to the BTB or the BHB.

Patterson and Hennessey, Figure 4.23 (reproduced below) shows:



The claim recites

setting a bit within an instruction text field of a branch, said bit preventing the branch from being placed into a branch history buffer and into a branch target buffer to thereby prevent the branch from being written into the branch history buffer and branch target buffer and predicted and to make the branch only detectable at the time frame of decode.

By way of contrast, the Figure jumps in and starts from a decision box "Entry found in branch target buffer" without any disclosure of how the system gets to the predecessor claimed state.

Then, the Final Action, citing Check, column 2, lines 28-31 (reproduced above), recites that

"... sensitive system operations require cache control, so for the same reason Check disables the BHT, the BTB would not need to be written to as well. Given the advantage of a BTB as disclosed by Patterson, and the need to implement it in the system as disclosed by Check, one of ordinary skill in the art at the time the invention was made would have been motivated to include a BTB, and also to disable its use when the BHT was disabled."

Applicants restate their position that this analysis and conclusion ignores the claim limitations

setting a bit within an instruction text field of a branch, said bit preventing the branch from being placed into a branch history buffer and into a branch target buffer to thereby prevent the branch from being written into the branch history buffer and branch target buffer and predicted and to make the branch only detectable at the time frame of decode

and is itself a hindsight reconstruction drawing from Applicants' disclosure and claims. Moreover, it is an incomplete hindsight reconstruction, the combination of references failing to specifically teach—

---"not use the BTB, as it would not have the data it needs to be made use of." and

---"include a BTB, and also to disable its use when the BHT was disabled."

But, as noted above, these are conclusions, hindsight reconstructions that are not based upon the four corners of the references, but upon Applicants' disclosure.

This argument is extended to independent claims 11 and 21, and to the claims dependent thereon (which carry all of the limitations of the independent claims from which they depend).

Conclusion

Based on the above discussion, it is respectfully submitted that the pending claims describe an invention that is statutory subject matter and is properly allowable to the Applicants.

If any issues remain unresolved despite the present amendment, the Examiner is requested to telephone Applicants' Attorney at the telephone number shown below to arrange for a telephonic interview before issuing another Office Action.

Applicants would like to take this opportunity to thank the Examiner for a thorough and competent examination and for courtesies extended to Applicants' Attorney.

Respectfully Submitted

Certificate of Electronic Filing

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Patent and Trademark Office on the date shown below.

Date of deposit: September 5, 2007

Person mailing paper: Richard M. Goldman

Signature: /s/ Richard M. Goldman

/s/ Richard M. Goldman

Richard M. Goldman, Reg. # 25,585
371 Elan Village Lane, Suite 208
San Jose, CA 95134
Voice: 408-324-0716
Fax: 408-324-0672
E-mail: goldmanptn@aol.com